



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/632,620	07/31/2003	Christopher Norman Kline	END920030057US1(1397-8U)	4750
68786	7590	12/27/2007	EXAMINER	
CHRISTOPHER & WEISBERG, P.A. 200 EAST LAS OLAS BOULEVARD SUITE 2040 FORT LAUDERDALE, FL 33301			KISS, ERIC B	
		ART UNIT	PAPER NUMBER	
		2192		
		MAIL DATE	DELIVERY MODE	
		12/27/2007	PAPER	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

MAILED  
DEC 27 2007

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

Application Number: 10/632,620

Filing Date: July 31, 2003

Appellant(s): KLINE, CHRISTOPHER NORMAN

Alan M. Weisberg (Reg. No. 43,982)  
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed November 19, 2007, appealing from the Office action mailed June 1, 2007.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

No amendment after final has been filed.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is substantially correct. However, upon further consideration some rejections have been withdrawn.

**WITHDRAWN REJECTIONS**

The following ground of rejection is not presented for review on appeal because it has been withdrawn by the examiner:

**The rejection of claims 25-31 under 35 U.S.C. § 101 is withdrawn.**

**REJECTIONS ON APPEAL**

Claims 21, 23-25, 27, and 28 remain rejected under 35 U.S.C. § 103(a) as being unpatentable over APA, Morrison, and OpenView.

Claims 22 and 26 remain rejected under 35 U.S.C. § 103(a) as being unpatentable over APA, Morrison, OpenView, and Boukobza.

Claims 29-31 remain rejected under 35 U.S.C. § 103(a) as being unpatentable over APA, Morrison, OpenView, and Wilson.

### **(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

### **(8) Evidence Relied Upon**

7,089,561	MORRISON et al.	08-2006
6,122,664	BOUKOBZA et al.	09-2000
2002/0166053	WILSON	11-2002

Nathan J. Muller, "Focus on OpenView A guide to Hewlett-Packard's Network and Systems Management Platform," 1995, pp. 1-291.

### **(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

**(A) Claims 21, 23-25 and 27-28** are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (hereinafter **APA**) in view of Morrison et al., US 7,089,561 (new art made of record & hereinafter **Morrison**) and further in view of Hewlett Packards "OpenView", by Nathan J. Muller (new art of record & hereinafter **OpenView**).

In regard to claim **21**, **APA** discloses:

- "*A computer implemented method for managing a change to a setting of a computer program of a computer system, said method comprising the steps of automatically attempting to change a setting of a computer program...*" (E.g., see "Background of the Invention", lines 18-19), wherein a systems administrator makes a change to an application via a prepared script.
- "*...said computer program writing a predetermined return code to a log file...and in response to step of automatically attempting to change said setting of said computer program...searching said log file for said return code...*"

(E.g., see “Background of the Invention”, lines 18-21), wherein a systems administrator verifies that a change was successful by checking output and certain codes and phrases in the output stored in temporary logs.

But, the **APA** does not expressly disclose “*...if said setting of said computer program is successfully changed, said computer program writing a predetermined return code to a log file, if said setting of said computer program is not successfully changed, said computer program not writing said return code to said log file, and in response to step of automatically attempting to change said setting of said computer program...*”. However, **Morrison** discloses:

- “*...if said setting of said computer program is successfully changed, said computer program writing a predetermined return code... if said setting of said computer program is not successfully changed, said computer program not writing said return code...*” (E.g., see Column 13), wherein return codes indicating success is taught if change is successful, otherwise a Win32 error.

**APA** and **Morrison** are analogous art because they are both concerned with the same field of endeavor, namely, implementing a setting change management method for a system. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Morrison**’s return codes with the **APA**’s change management’s log file to provide a log file with a successful return code. The motivation to do so would have been to verify that the change was made successfully by reviewing the output of those commands in the log file as taught by the **APA** (E.g., see APA, page 1, lines 21-25).

But, the **APA** or **Morrison** do not expressly disclose automatically sending a notification regarding whether the setting of the computer program was successfully changed or not. However, **OpenView** discloses:

- automatically sending a notification regarding whether the setting of the computer program was successfully changed or not (see, e.g., p. 32 and p. 46

(“Event Reporting”)), wherein an administrator sets a schedule for an automatic installation.

**APA, Morrison** and **Openview** are analogous art because they are both concerned with the same field of endeavor, namely, implementing a setting change management method for a system. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Morrison** and **APA**’s logged return code status with **Openview**’s teaching of notification to an administer. The motivation to do so would have been to provide comprehensive verifications of installed software (change) to an administrator (E.g., see OpenView, page 182, fourth paragraph).

In regard to claim 23, the rejections of base claim 21 are incorporated. Furthermore, **OpenView** discloses:

- “*...sent to an administrator.*” (E.g., OpenView, page 32, first paragraph), wherein an administrator is alerted by a pre-specified action on event which involves sending an audio alert when specified events occur.

In regard to claim 24, the rejections of base claim 21 are incorporated.

Furthermore, **APA** discloses:

- “*...said return code is a response to a command to change said setting.*” (E.g., see “Background”, lines 21-25), wherein a command or script issue commands to change permissions is taught.

In regard to claims 25, 27-28, this is a computer program product version of the claimed method discussed above, in claims 21, 23-24, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see **OpenView** wherein a computer network, system and program products are taught (e.g., see Chapter 1).

(B) Claims 22 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over **APA** in view of **Morrison** and further in view of **OpenView** in view of Boukobza et al., US 6,122,664 (new art of record & hereinafter **Boukobza**).

In regard to claim 22, the rejections of base claim 21 are incorporated. But, the **APA, Morrison or Openview** do not expressly disclose “*...said setting is a number of messages that can reside on a queue used by said computer program*”. However, **Boukobza** discloses:

“*...said setting is a number of messages that can reside on a queue...*” (E.g., see Column 19, lines 20-52), wherein creating a new number-of-messages parameter in the queue is taught.

**APA, Morrison, Openview** and **Boukobza** are analogous art because they are both concerned with the same field of endeavor, namely, a method for monitoring data structures in a computer system. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Morrison, APA** and **Openview**’s method of monitoring change with **Boukobza**’s setting of a number of messages residing on a queue. The motivation to do so would have been to provide a method to test whether the “log” file is growing abnormally as taught by **Boukobza** (E.g., see Column 20, lines 40-50).

In regard to claim 26 see claim 22.

(C) Claims 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **APA** in view of **Morrison** and further in view of **OpenView** in view **Wilson**, US 2002/0166053 (new art of record & hereinafter **Wilson**).

In regard to claim 29, this is another computer program product comprising the claimed method discussed above, in claims 21 and 24. Furthermore, **APA** discloses:

“*...a permission file...*” (E.g., see “Background”, lines 21-25), wherein a command or script issues commands to change permissions is taught.

But, the **APA, Morrison** or **OpenView** do not expressly disclose “*...said permission file indicating which User Ids have access to which other files...*”.

However, **Wilson** discloses:

“*...said permission file indicating which User Ids have access to which other files...*” (E.g., see Figure 2 & paragraph [0024]), wherein a permission file metadata is taught.

**APA, Morrison, OpenView** and **Wilson** are analogous art because they are both concerned with the same field of endeavor, namely, a method for managing a computer network comprising permission files. Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Wilson**’s permission files with the **APA, Morrison** and **Openview**’s method of monitoring change. The motivation to do so would have been to manage the configuration (E.g., see OpenView, page 36) in a computer network system.

In regard to claim 30, the rejections of base claim 29 are incorporated. Furthermore, **APA** discloses:

“*...permission file is part of the operating system.*” (E.g., see “Background”, lines 21-23), wherein the script to adjust permission files is part of the operating system.

In regard to claim 31 see claim 23.

#### (10) Response to Argument

##### The Rejection of Claims 25-31 under 35 U.S.C. § 101 (Brief at 4-5)

Appellant's arguments are moot in view of the withdrawn rejection.

##### The Rejection of Claims 21, 23-25, 27, and 28 under 35 U.S.C. § 103(a) (Brief at 5-9)

Appellant does not dispute that the "Background" section of the disclosure contains admitted prior art. Rather, appellant appears to dispute the *extent* of the admission and the implications of such an admission with regard to obviousness under § 103(a). (Brief at 6.)

Appellant asserts that the Background section, "discloses *manually* checking for the presence of an output after *manually* entering commands to initiate a change in a computer program." (*Id.* (emphasis in original).) However, a careful review of the Background section shows that more is admitted than appellant lets on. In addition to the manual entry of commands to initiate a change in a computer program, the Background section in several places also clearly admits that it has been known to initiate a change in a computer program by automatic means:

Heretofore, the systems administrators have made [configuration and other changes . . .] by manually entering commands or initiating scripts in real time to implement the changes.

For example, to create a new queue, the systems administrator has entered commands or initiated a script which identified an application instance which is the target of the change, and issued change commands via an application interface.

The systems administrator then made a change to the application instance, either manually or via a prepared script . . .

As another example of how a systems administrator made a change in real time to adjust permissions, the systems administrator has entered commands or initiated a script which issued operating system commands.

(Specification at 1 (emphasis added); *see also* Specification at 2 (“The systems administrators have run [other types of jobs] by manually entering commands or initiating scripts in real time to implement the change.”) (emphasis added).) The initiation of scripts to implement changes, explicitly disclosed as an admittedly known alternative to manually entering commands to implement such changes, may be reasonably interpreted as, “automatically attempting to change a setting of a computer program,” in the context of claims 21 and 25.

Further, the Background section readily admits that it has been known to manually search a log file in response to [manually or automatically] attempting to change a setting of a computer program. (Specification at 1 (“The systems administrator then made a change to the application instance, either manually or via a prepared script, and verified that the change was made successfully by manually checking the output of the change and searching for certain codes and phrases in output either displayed on the screen or stored in temporary logs.”) (emphasis added).) Thus, what is missing in regard to the language of claims 21 and 25 is the automation of this admittedly known manual process of searching a log file in response to [manually or automatically] attempting to change a setting of a computer program. However, broadly providing an automatic or mechanical means to replace a manual activity which accomplished the same result is not sufficient to distinguish over the prior art. *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194 (CCPA 1958).

In further support of the § 103(a) rejection, the examiner cited OpenView as teaching that it has been known to generate automatic notifications regarding the output of configuration change commands (known to be stored in log files). *See, e.g.*, OpenView at pp. 6-8 (describing the Event Management and Data Management Services), pp. 27-28 (describing the Event

Manager), p. 36 (describing Configuration Management), p. 36 (describing Event Reporting, including events for changes in a node's configuration or occurrences of application alerts), pp. 90-91 (describing the notification of success or failure of an automatic action), pp. 183-185 (describing Admincenter, which provides automatic change management capabilities).

In view of the full extent of the admitted prior art contained in the Background section of the specification, the limited nature of the missing features (*i.e.*, merely automating a known manual activity which accomplished the same result), and the cited evidence that such automation has likewise been known in the prior art, the examiner respectfully submits that the rejection of claims 21, 23-25, 27, and 28 under 35 U.S.C. § 103(a) should be affirmed.

**The Rejection of Claims 22 and 26 under 35 U.S.C. § 103(a) (Brief at 9)**

Appellant states that claims 22 and 26 stand or fall together with claims 21 and 25. (Brief at 9.) Therefore, for at least the reasons set forth above with regard to claims 21 and 25, the examiner respectfully submits that the rejection of claims 22 and 26 should be affirmed.

**The Rejection of Claims 29-31 under 35 U.S.C. § 103(a) (Brief at 9-11)**

Again (see the response to appellant's arguments regarding claims 21 and 25 above), appellant asserts that the Background section, "discloses the prior art required *manually* checking for the presence of an output after *manually* entering commands to initiate a change in a computer program . . . ." (Brief at 10 (emphasis in original).) Again, this assertion is only partly true. As noted above, the Background section clearly admits that it has been known to initiate a change in a computer program by automatic means:

Heretofore, the systems administrators have made [configuration and other changes . . . ] by manually entering commands or initiating scripts in real time to implement the changes.

....  
For example, to create a new queue, the systems administrator has entered commands or initiated a script which identified an application instance which is the target of the change, and issued change commands via an application interface.

....  
The systems administrator then made a change to the application instance, either manually or via a prepared script . . . .

....  
As another example of how a systems administrator made a change in real time to adjust permissions, the systems administrator has entered commands or initiated a script which issued operating system commands.

(Specification at 1 (emphasis added); *see also* Specification at 2 (“The systems administrators have run [other types of jobs] by manually entering commands or initiating scripts in real time to implement the change.”) (emphasis added).)

Regarding the attempted changes in a permission file indicating which User Ids have access to which other files, the Background section further admits that it has been known to automatically attempt these types of changes.

Systems administrators are often required to make configuration or other changes to computer systems, such as . . . adjusting permissions for accessing files . . . .  
Heretofore, the systems administrators have made such changes by manually entering commands or initiating scripts in real time to implement the changes.

....  
As another example of how a systems administrator made a change in real time to adjust permissions, the systems administrator has entered commands or initiated a script which issued operating system commands. The systems administrator verified that the change was made successfully by reviewing the output of those commands, as well as the exit status of each command.

(Specification at 1 (emphasis added).) The initiation of scripts to implement changes to permissions, explicitly disclosed as an admittedly known alternative to manually entering

commands to implement such changes, may be reasonably interpreted as, “automatically attempting to change a permission file,” in the context of claim 29.

Further, the Background section readily admits that it has been known to manually search a log file in response to [manually or automatically] attempting to change a setting of a computer program.

The systems administrator then made a change to the application instance, either manually or via a prepared script, and verified that the change was made successfully by manually checking the output of the change and searching for certain codes and phrases in output either displayed on the screen or stored in temporary logs.

....  
The systems administrator verified that the change was made successfully by reviewing the output of those commands, as well as the exit status of each command.

(Specification at 1 (emphasis added).) Thus, what is missing in regard to the language of claim 29 is the automation of this admittedly known manual process of searching a log file in response to [manually or automatically] attempting to change a setting of a computer program. However, broadly providing an automatic or mechanical means to replace a manual activity which accomplished the same result is not sufficient to distinguish over the prior art. *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194 (CCPA 1958).

In further support of the § 103(a) rejection, the examiner cited OpenView as teaching that it has been known to generate automatic notifications regarding the output of configuration change commands (known to be stored in log files). *See, e.g.*, OpenView at pp. 6-8 (describing the Event Management and Data Management Services), pp. 27-28 (describing the Event Manager), p. 36 (describing Configuration Management), p. 36 (describing Event Reporting, including events for changes in a node’s configuration or occurrences of application alerts), pp.

90-91 (describing the notification of success or failure of an automatic action), pp. 183-185 (describing Admincenter, which provides automatic change management capabilities).

In view of the full extent of the admitted prior art contained in the Background section of the specification, the limited nature of the missing features (*i.e.*, merely automating a known manual activity which accomplished the same result), and the cited evidence that such automation has likewise been known in the prior art, the examiner respectfully submits that the rejection of claims 29-31 under 35 U.S.C. § 103(a) should be affirmed.

#### **(11) Related Proceeding(s) Appendix**

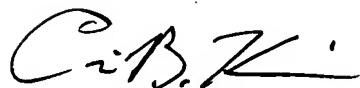
No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Application/Control Number:  
10/632,620  
Art Unit: 2192

Page 15

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,



Eric B. Kiss

Primary Patent Examiner, Art Unit 2192

Conferees:



TUAN DAM  
SUPERVISORY PATENT EXAMINER

Tuan Q. Dam

Supervisory Patent Examiner, Art Unit 2192

Wei Zhen

Supervisory Patent Examiner, Art Unit 2191



WEI ZHEN  
SUPERVISORY PATENT EXAMINER